

# Over-constrained Weierstrass iteration and the nearest consistent system

Olivier Ruatta\*, Mark Sciabica, Agnes Szanto†

January 22, 2014

## Abstract

We propose a generalization of the Weierstrass iteration for over-constrained systems of equations and we prove that the proposed method is the Gauss-Newton iteration to find the nearest system which has at least  $k$  common roots and which is obtained via a perturbation of prescribed structure. In the univariate case we show the connection of our method to the optimization problem formulated by Karmarkar and Lakshman for the nearest GCD. In the multivariate case we generalize the expressions of Karmarkar and Lakshman, and give explicitly several iteration functions to compute the optimum. The arithmetic complexity of the iterations is detailed.

**Keywords:** Overdetermined systems, nearest consistent system, Weierstrass Durand Kerner method

## 1 Introduction

In many physical and engineering applications one needs to solve over-constrained systems of equations, i.e. systems with more equations than unknowns, such that the existence of the solutions is guaranteed by some underlying physical property. However, the input system may be given only with limited accuracy due to measurement or rounding error, and thus the actual input may be inconsistent.

The work presented in this paper is concerned with the question of finding the “nearest” system with at least  $k$  distinct common roots over  $\mathbb{C}$ . We introduce a generalization of the Gauss-Weierstrass method [38, 31]. In the univariate case, the proposed iterative method allows computation of the nearest GCD of given degree, and is closely related to the formula of Karmarkar-Lakshman for the distance to the set of systems with at least  $k$  common roots [22, 23]. We show how to extend the iterative method to over-constrained systems of analytic functions. Using this

---

\*Université de Limoges

†North Carolina State University, Raleigh, NC. This research was partly supported by NSF grants CCR-0306406, CCR-0347506, DMS-0532140 and CCF-1217557.

extended construction we generalize the Karmarkar-Lakshman formula to the multivariate case.

More precisely, in the univariate case the problem we address in the paper is the following:

**Problem 1** *Given  $f, g \in \mathbb{C}[x]$  and  $k \in \mathbb{N}$ , find a polynomial  $h$  of degree  $k$  such that there exist polynomials  $\tilde{f}, \tilde{g} \in \mathbb{C}[x]$  such that  $h$  divides both  $\tilde{f}$  and  $\tilde{g}$ , and  $f - \tilde{f}$  and  $g - \tilde{g}$  have prescribed supports and minimal 2-norms.*

The method proposed here is based on a generalization of the so-called Weierstrass method (also called Durand-Kerner method [9, 24, 10] or Dochev method [14, 35]) introduced in [38] and successively generalized in [2, 30, 31, 27] (for a survey on the history see [28]). Our first contribution in the univariate case is to show a link between the Weierstrass method and the formulation of Karmarkar and Lakshman in [23] using Lagrange interpolation (see Theorem 2.7). The second contribution is an explicit formula for the Gauss-Newton iteration to find the optimum, which is derived from our expressions for the gradient of the norm square function (see Theorem 2.11).

Next we present the extension of our results to the multivariate case. The problem we address is as follows:

**Problem 2** *Given an analytic function  $\vec{f} = (f_1, \dots, f_N) : \mathbb{C}^n \rightarrow \mathbb{C}^N$ ,  $N > n$ , and  $k > 0$ , find perturbations  $p_1, \dots, p_N$  from a given finite dimensional vector space  $\mathcal{P}$  of analytic functions together with  $k$  distinct points  $\mathbf{z}_1, \dots, \mathbf{z}_k \in \mathbb{C}^n$ , such that  $f_1 - p_1, \dots, f_N - p_N$  vanishes on  $\mathbf{z}_1, \dots, \mathbf{z}_k$  and  $\|p_1\|_2^2 + \dots + \|p_N\|_2^2$  is minimal.*

One of the main results of the paper is a generalization of the formula of Karmarkar and Lakshman in [23] for the univariate nearest GCD to the multivariate case. Using a generalization of the Lagrange interpolation we were able to express the distance of our input system to the set of systems which have at least  $k$  complex roots as an optimization problem on the  $k$ -tuples of points in  $\mathbb{C}^n$  (see Theorem 3.7). The other main result of the paper is an explicit formulation of the Gauss-Newton iteration applied to our optimization formulation to solve Problem 2.

Finally, we give a simplified version of the iteration, which might be of independent interest. Analogously to the classical Weierstrass map, we use the multivariate Lagrange interpolation polynomials in each Gauss-Newton iteration step to transform the Jacobian matrix to a block diagonal matrix. As a consequence, we get a simple component-wise formula for the iteration function. We show that using the simplified method, the complexity of computing each iterate is improved compared to the non-simplified versions: the standard Gauss-Newton iteration, the quadratic iteration, or the conjugate gradient method. However, the simplified iteration will not converge to the least squares minimum, but we do give a description of its fixed points. As our numerical experiments indicate, the simplified method computes roots with the smallest residual value  $\sum_{i=1}^N \sum_{j=1}^k |f_i(\mathbf{z}_j)|^2$ , compared to the non-simplified versions.

At the end of the paper we present numerical experimentations where we compare the performances of the simplified Gauss-Newton, the standard Gauss-Newton, the quadratic iteration, and the conjugate gradient methods to compute the optimum.

## 1.1 Related work

The computation of the GCD is a classical problem of symbolic computation and efficient algorithms are known to solve it ([5] and [3] for instance). The first approach to a problem similar to Problem 1 was proposed by Schönhage in [33] where the input polynomials are known with infinite precision. Several later approaches were proposed where the polynomials are known with a bounded error. In [12, 6] the authors compute upper bounds on the degree of an  $\epsilon$ -GCD of two numerical polynomials using the singular values of a Sylvester resultant matrix. In [13], the authors give the exact degree of the  $\epsilon$ -GCD together with a certificate using a singular value decomposition of a subresultant matrix. In [22, 23, 4], the authors present the problem as a real optimization problem and propose numerical techniques in order to solve the optimization problem. Hitz et al. consider the nearest polynomial with constrained or real roots in the  $l^2$  and  $l^\infty$  norms in [18, 19]. Related approaches on approximate GCD computation include [32, 49, 46, 37, 48, 47, 29, 20, 21, 34, 40, 39, 25, 41, 44, 43, 11, 42].

There are two main families of approaches in the literature to compute the solution of multivariate near-consistent over-constrained systems. One type of algorithm handles over-constrained polynomial systems with approximate coefficients by using a symbolic-numeric approach to reduce the problem to eigenvalue computation via *multiplication tables*. The first methods in the literature using reduction to eigenvalue problem include [1, 45, 26]. The existing methods to compute the multiplication tables use resultant matrices or Gröbner basis techniques, with complexity bound exponential in the number of variables.

The other type of approaches formulate over-constrained systems as real optimization problems. Here we can only list a selected subset of the related literature. Giusti and Schost in [15] reduce the problem to the solution of a univariate polynomial. Dedieu and Shub give a heuristic predictor corrector method in [7]. They also prove alpha-theory for the Gauss-Newton method in [8]. Stetter in [36] studies the conditioning properties of near-consistent over-constrained systems. Ruatta in [31] generalizes the Weierstrass iteration for over-constrained systems and gives a heuristic predictor corrector method based on this iteration. Recently, Hauenstein and Sottile considered certification of approximate solutions of exact overdetermined systems in [17].

## 1.2 Notations

In all that follows,  $\mathbb{C}$  denotes the field of complex numbers,  $x$  is an indeterminate and we denote by  $\mathbf{x} = (x_1, \dots, x_n)$  the vector of  $n$  indeterminates for some  $n \geq 1$ .  $\mathbb{C}[x]$  and  $\mathbb{C}[\mathbf{x}]$  denote the rings of polynomials with complex coefficients in one and  $n$  indeterminates, respectively.  $\mathbb{C}[x]_m$  is the subspace of  $\mathbb{C}[x]$  consisting of the polynomials of degree less or equal to  $m \in \mathbb{N}$ . For  $I \subset \mathbb{N}$  a finite

set, we denote  $\mathbb{C}[x]_I$  the set of polynomials with support included in  $I$ , i.e.

$$\mathbb{C}[x]_I = \{p \in \mathbb{C}[x] : p(x) = \sum_{i \in I} p_i x^i, p_i \in \mathbb{C}\}. \quad (1)$$

For  $F \subset \mathbb{C}[\mathbf{x}]$  and  $\mathcal{R} \subseteq \mathbb{C}^n$  we denote by  $\mathbf{V}_{\mathcal{R}}(F)$  the set of common roots of  $F$  in  $\mathcal{R}$ . We denote indifferently  $\|\cdot\|_2$  or  $\|\cdot\|$  the  $l^2$  norm of complex vectors which we call the 2-norm. For  $f \in \mathbb{C}[x]$  we denote by  $\|f\|$  the 2-norm of the vector of its coefficients. For  $M \in \mathbb{C}^{k \times m}$ ,  $\|M\|$  denotes the 2-norm of the vector of its entries. The 2-norm of a vector of polynomials is the 2-norm of the vector of all their coefficients. For a matrix  $M \in \mathbb{C}^{k \times n}$  we denote by  $M^T$  its transpose matrix and  $M^*$  the transpose of the conjugate of  $M$ , also called the adjoint of  $M$ . For  $M \in \mathbb{C}^{k \times m}$  such that  $\text{rank}(M) = k$  (or  $\text{rank}(M) = m$ ), we denote by  $M^\dagger = M^*(MM^*)^{-1}$  (or  $M^\dagger = (M^*M)^{-1}M^*$ , respectively) its Moore-Penrose pseudo-inverse.

## 2 Univariate case

In this section, we present a generalization of the Weierstrass iteration to the approximate case. First we present a version of the classical Lagrange interpolation method which is needed for the construction of the iterative method. Secondly, we define a generalization of the Weierstrass map and show the link between this map and the distance to the set of systems with  $k$  common roots, translating this distance from a minimization problem on the coefficient vector of the perturbations to a minimization problem over  $k$ -tuples of complex number. Next we give an explicit formula for the Gauss-Newton iteration for our optimization formulation. Finally, we give a simplified version of the iteration, which has a simple coordinate-wise iteration function with improved complexity.

### 2.1 Generalized Lagrange interpolation

In this subsection we introduce an optimization problem which generalizes the classical Lagrange interpolation problem and we give a solution to this problem using Moore-Penrose pseudo-inverses.

**Problem** [Generalized Lagrange interpolation] *Consider distinct complex numbers  $z_1, \dots, z_k \in \mathbb{C}$  and some arbitrary complex numbers  $f_1, \dots, f_k \in \mathbb{C}$ . Fix  $I \subset \mathbb{N}$  such that  $|I| \geq k$ . The generalized Lagrange interpolation problem consists of finding the minimal 2-norm polynomial  $F \in \mathbb{C}[x]_I$  with support  $I$  that satisfies:*

$$F(z_i) = f_i \text{ for } i = 1, \dots, k. \quad (2)$$

We will need the following definition:

**Definitions 2.1** *Let  $I = \{i_1, \dots, i_p\} \subset \mathbb{N}$  such that  $p \geq k$ .*

- *Let  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{C}^k$ . We define the Vandermonde matrix associated with  $\mathbf{z}$  and  $I$  as following matrix of size  $k \times p$ :*

$$V_I(\mathbf{z}) := \begin{pmatrix} z_1^{i_1} & \dots & z_1^{i_p} \\ \vdots & \ddots & \vdots \\ z_k^{i_1} & \dots & z_k^{i_p} \end{pmatrix}. \quad (3)$$

- For  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{C}^k$  we define the  $k \times k$  matrix  $M_I(\mathbf{z})$  by:

$$M_I(\mathbf{z}) = \left( \sum_{i \in I} (z_s \bar{z}_t)^i \right)_{s,t=1,\dots,k}. \quad (4)$$

Note that  $M_I(\mathbf{z}) = V_I(\mathbf{z})^* V_I(\mathbf{z})$ .

- For  $I \subset \mathbb{N}$  we define  $\mathcal{R}_I := \{(z_1, \dots, z_k) \in \mathbb{C}^k \mid \text{rank}(V_I(\mathbf{z})) = k\}$ . For  $I, J \subset \mathbb{N}$  we define  $\mathcal{R}_{I,J} := \mathcal{R}_I \cap \mathcal{R}_J$ .
- For  $I, J \subset \mathbb{N}$  and  $f, g \in \mathbb{C}[x]$  we define the set

$$\Omega_{I,J,k}(f, g) := \left\{ (\tilde{f}, \tilde{g}) \mid \exists (z_1, \dots, z_k) \in \mathcal{R}_{I,J} \forall i \tilde{f}(z_i) = \tilde{g}(z_i) = 0; f - \tilde{f} \in \mathbb{C}[x]_I, g - \tilde{g} \in \mathbb{C}[x]_J \right\}.$$

Informally,  $\Omega_{I,J,k}(f, g)$  is the set of pairs with at least  $k$  common roots which are obtained from  $(f, g)$  via perturbation of the coefficients corresponding to  $I$  and  $J$ , respectively. We may omit  $(f, g)$  from  $\Omega_{I,J,k}(f, g)$  if it is clear from the context.

Next we introduce a family of polynomials which can be viewed as the generalization of the Lagrange polynomials.

**Definition 2.2** Let  $\mathbf{z} \in \mathcal{R}_I$  and let  $V_I(\mathbf{z})$  be the generalized Vandermonde matrix associated with  $\mathbf{z}$  and  $I$ . Define  $\mathbf{x}_I = (x^{i_1}, \dots, x^{i_p})$  and denote by  $\{\mathbf{e}_1, \dots, \mathbf{e}_k\} \subset \mathbb{C}^k$  the standard basis of  $\mathbb{C}^k$ . We define the **generalized Lagrange polynomials** with support in  $I$  as follows:

$$L_{I,i}(\mathbf{z}, x) := \mathbf{x}_I V_I(\mathbf{z})^\dagger \mathbf{e}_i \quad i = 1, \dots, k. \quad (5)$$

Note that if  $I = \{0, \dots, k-1\}$  then  $\{L_{I,i}(\mathbf{z}, x) \mid 1 \leq i \leq k\}$  are the classical Lagrange interpolation polynomials.

The following propositions assert that the generalized Lagrange polynomials allow us to find the minimal norm polynomial with prescribed support  $I$  satisfying (2). We also highlight the connection between the 2-norms of the interpolation polynomials and the results of Karmarkar and Lakshman in [23].

**Proposition 2.3** Let  $I \subset \mathbb{N}$  with  $p \geq k$  and  $\mathbf{z} = (z_1, \dots, z_k) \in \mathcal{R}_I$ . Then for all  $1 \leq i, j \leq k$ ,  $L_{I,i}(\mathbf{z}, z_j) = \delta_{i,j}$ .

**Proof** From (5) we get that  $L_{I,i}(\mathbf{z}, z_j) = \mathbf{e}_j^T V_I(\mathbf{z}) V_I(\mathbf{z})^\dagger \mathbf{e}_i$  for all  $i, j \in \{1, \dots, k\}$ . Then we use that  $V_I(\mathbf{z})$  has rank  $k$  to get that  $V_I(\mathbf{z})^\dagger$  is the right inverse of  $V_I(\mathbf{z})$ , thus  $V_I(\mathbf{z}) V_I(\mathbf{z})^\dagger = id$ .  $\square$

**Proposition 2.4** Let  $I \subset \mathbb{N}$ ,  $\mathbf{z} \in \mathcal{R}_I$  and  $\mathbf{f} = (f_1, \dots, f_k) \in \mathbb{C}^k$ . Define

$$F(x) := \sum_{i=0}^k f_i L_{I,i}(\mathbf{z}, x). \quad (6)$$

Then we have  $F(x) \in \mathbb{C}[x]_I$  and

$$F(z_j) = f_j, \forall j \in \{1, \dots, k\}. \quad (7)$$

Moreover,

$$\|F\|^2 = \mathbf{f}^* M_I(\mathbf{z})^{-1} \mathbf{f} \quad (8)$$

is minimal among the polynomials in  $\mathbb{C}[x]_I$  satisfying (7).

**Proof** Let  $F(x)$  be as in (6). If we denote by  $\mathbf{F} = (F_i)_{i \in I}$  the vector of coefficients of  $F(x)$  then by the definition of the generalized Lagrange polynomials we have  $\mathbf{F} = V_I(\mathbf{z})^\dagger \mathbf{f}$ . It is easy to check that  $\|\mathbf{F}\|^2 = \mathbf{f}^* M_I^{-1} \mathbf{f}$  using the fact that  $M_I^{-1} = V_I(\mathbf{z})^{+*} V_I(\mathbf{z})^\dagger$ . On the other hand,  $\mathbf{F}$  is the minimal 2-norm vector satisfying  $V_I(\mathbf{z})\mathbf{F} = \mathbf{f}$ , which follows from the properties of the Moore-Penrose pseudo-inverse (see [16]). Finally, we note that  $V_I(\mathbf{z})\mathbf{F} = \mathbf{f}$  is equivalent to (7).  $\square$

The above propositions allow us to state the main result of the subsection:

**Theorem 2.5** *Let  $f, g \in \mathbb{C}[x]$ ,  $I, J \subset \mathbb{N}$  and  $\mathbf{z} \in \mathcal{R}_{I,J}$ . We define the following polynomials in  $\mathbb{C}[x]_I$  and  $\mathbb{C}[x]_J$ , respectively:*

$$F_I(\mathbf{z}, x) := \sum_{i=1}^k f(z_i) L_{I,i}(\mathbf{z}, x), \quad G_J(\mathbf{z}, x) := \sum_{i=1}^k g(z_i) L_{J,i}(\mathbf{z}, x). \quad (9)$$

Then

$$(f(x) - F_I(\mathbf{z}, x), g(x) - G_J(\mathbf{z}, x)) \in \Omega_{I,J,k}(f, g).$$

Moreover, if  $\min_{\mathbf{z} \in \mathcal{R}_{I,J}} \left( \mathbf{f}^* M_I(\mathbf{z})^{-1} \mathbf{f} + \mathbf{g}^* M_J(\mathbf{z})^{-1} \mathbf{g} \right)$  exists and is reached at  $\zeta \in \mathcal{R}_{I,J}$  then we have

$$\|F_I(\zeta, x)\|^2 + \|G_J(\zeta, x)\|^2 = \min_{(\tilde{f}, \tilde{g}) \in \Omega_{I,J,k}} \left\{ \|f - \tilde{f}\|^2 + \|g - \tilde{g}\|^2 \right\}.$$

Here  $\mathbf{f} = (f(z_1), \dots, f(z_k)) \in \mathbb{C}^k$  and  $\mathbf{g} = (g(z_1), \dots, g(z_k)) \in \mathbb{C}^k$ .

**Proof** The proof can be deduced easily from the proposition 2.4.  $\square$

## 2.2 Generalized Weierstrass map

In this section we give a generalization of the univariate over-constrained Weierstrass map introduced in [31]. Informally, for  $f, g \in \mathbb{C}[x]$  the Weierstrass map  $\mathcal{W}$  in [31] is a map defined on  $\mathbb{C}^k$  with the property that  $\mathcal{W}(z_1, \dots, z_k) = 0$  if and only if  $f(z_i) = g(z_i) = 0$  for  $1 \leq i \leq k$ . The main contribution of this subsection is the observation that the norm  $\|\mathcal{W}(\mathbf{z})\|_2$  is closely related to the distance defined by Karmarkar and Lakshman in [23]. Using this observation, it is straightforward to see that the least square minimum of the Weierstrass map  $\mathcal{W}$  corresponds to the  $k$  common roots of the closest system  $\tilde{f}, \tilde{g}$  which is obtained from  $f, g$  via the perturbation of a prescribed

subset of their coefficients.

First we give the definition of the generalized Weierstrass map using the generalized Lagrange polynomials defined in (5).

**Definition 2.6** Let  $f, g \in \mathbb{C}[x]$ ,  $k \geq 1$  and  $I, J \subset \mathbb{N}$  such that  $|I|, |J| \geq k$ . For a fixed  $\mathbf{z} \in \mathcal{R}_{I,J}$ , let  $F_I(\mathbf{z}, x) \in \mathbb{C}[x]_I$  and  $G_J(\mathbf{z}, x) \in \mathbb{C}[x]_J$  be the interpolation polynomials defined in (9). Then the map defined by

$$\mathcal{W}_{I,J} : \begin{cases} \mathbb{C}^k \rightarrow \mathbb{C}[x]_I \oplus \mathbb{C}[x]_J \\ \mathbf{z} \mapsto (F_I(\mathbf{z}, x), G_J(\mathbf{z}, x)) \end{cases} \quad (10)$$

is called the **generalized Weierstrass map** with supports  $I$  and  $J$ .

In the next theorem we prove that the least square solution of the Weierstrass map and the optimization problem posed by Karmarkar and Lakshman in [23] are closely related.

**Theorem 2.7** Let  $\mathbf{z} = (z_1, \dots, z_k)$ ,  $(f, g)$ , and  $\mathcal{W}_{I,J}$  be as in Definition 2.6. Then

- i.  $\mathcal{W}_{I,J}(\mathbf{z}) = 0$  if and only if  $(z_1, \dots, z_k)$  are common roots of  $f$  and  $g$ .
- ii. Using the notation of Theorem 2.5, for all  $\mathbf{z} \in \mathbb{C}^k$  we have

$$\|\mathcal{W}_{I,J}(\mathbf{z})\|^2 = \mathbf{f}^* M_I^{-1} \mathbf{f} + \mathbf{g}^* M_J^{-1} \mathbf{g}.$$

$$\text{iii. } \min_{\mathbf{z} \in \mathcal{R}_{I,J}} \|\mathcal{W}_{I,J}(\mathbf{z})\|^2 = \min_{(\tilde{f}, \tilde{g}) \in \Omega_{I,J,k}} \left\{ \|f - \tilde{f}\|^2 + \|g - \tilde{g}\|^2 \right\}.$$

**Proof** (i)  $\mathcal{W}_{I,J}(\mathbf{z}) = 0$  if and only if  $F_I(\mathbf{z}, x) = G_J(\mathbf{z}, x) = 0$  for all  $x \in \mathbb{C}$ . This implies that  $f(z_i) = F_I(\mathbf{z}, z_i) = 0$  and  $g(z_i) = G_J(\mathbf{z}, z_i) = 0$  for all  $1 \leq i \leq k$ . On the other hand, assume that  $z_1, \dots, z_k$  are common roots of  $f$  and  $g$ . Since  $F_I$  and  $G_J$  are the minimal 2-norm polynomials interpolating  $(f(z_1), \dots, f(z_k)) = 0$  and  $(g(z_1), \dots, g(z_k)) = 0$ ,  $F_I$  and  $G_J$  must both be the zero polynomial.

(ii) follows from the definition of  $\mathcal{W}_{I,J}$  in (10), the definition of  $F_I(\mathbf{z}, x)$  and  $G_J(\mathbf{z}, x)$  in (9) and from (8).

(iii) follows from (ii) and from Theorem 2.5.  $\square$

**Remark 2.8** As a special case of the above proposition, we get that the least squares solution of the univariate over-constrained Weierstrass map  $\mathcal{W}$  defined in [31] gives the common roots of the closest system with  $k$  common roots, and obtained via the perturbation of the coefficients corresponding to  $I = J = \{0, 1, \dots, k-1\}$ , i.e. the terms of  $f$  and  $g$  of degree less than  $k$ . This gives a link between the Weierstrass map of [31] and the distance formulated for the approximate GCD problem by Karmarkar and Lakshman in [23].

### 2.3 Gauss-Newton iteration

In Theorem 2.7 we obtained a formulation for the distance of  $f, g$  from the set of pairs with at least  $k$  common roots as the 2-norm minimum of the Weierstrass map  $\mathcal{W}_{I,J}$ . In this subsection we give explicit formulas for the Gauss-Newton iteration for  $\mathcal{W}_{I,J}$ . The theoretical framework for the Gauss-Newton iteration for computing the 2-norm optimum of complex functions is described in the multivariate setting in Section 4, in the present subsection we present our results without proof.

First we would like to note that if  $|I| \neq k$  or  $|J| \neq k$  then the function  $\mathcal{W}_{I,J}(\mathbf{z})$  is not a complex analytic function. However, in this case we can separate the original complex variables  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{C}^k$  and their conjugate  $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_k) \in \mathbb{C}^k$ , and express  $\mathcal{W}_{I,J}$  as a function of both of them. A simple computation described in Section 4 shows that the vanishing of the gradient of  $\|\mathcal{W}_{I,J}\|^2$  will result in two equations which are conjugates of each other. Thus solving only one of them will result to the definition of the Gauss-Newton iteration as follows (see more details in Section 4):

$$\mathbf{z}^{new} = \mathbf{z} - \mathcal{J}(\mathbf{z})^\dagger \mathcal{W}_{I,J}(\mathbf{z}), \quad (11)$$

where  $\mathcal{J}(\mathbf{z})$  is the Jacobian matrix of  $\mathcal{W}_{I,J}$  at  $\mathbf{z}$  of size  $(|I| + |J|) \times k$ .

In the following proposition we give an expression of the Gauss-Newton iteration computed by conducting linear algebra on the Vandermonde matrices  $V_I(\mathbf{z})$  and  $V_J(\mathbf{z})$ .

**Proposition 2.9** *Let  $f, g \in \mathbb{C}[x]$ ,  $k > 0$ , and  $I, J \subset \mathbb{N}$  such that  $|I|, |J| \geq k$ . For a fixed  $k$ -tuple  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{C}^k$  of distinct numbers define*

$$f_{\mathbf{z}}(x) := f - F_I(\mathbf{z}, x) \text{ and } g_{\mathbf{z}} := g - G_J(\mathbf{z}, x) \quad (12)$$

using (9). The iteration defined by

$$\mathbf{z}^{new} := \mathbf{z} - (D_{f_{\mathbf{z}}}^* M_I^{-1} D_{f_{\mathbf{z}}} + D_{g_{\mathbf{z}}}^* M_J^{-1} D_{g_{\mathbf{z}}})^{-1} (D_{f_{\mathbf{z}}}^* M_I^{-1} \mathbf{f} + D_{g_{\mathbf{z}}}^* M_J^{-1} \mathbf{g}) \quad (13)$$

is the Gauss-Newton iteration for the Weierstrass map  $\mathcal{W}_{I,J}$ . Here

$$D_{f_{\mathbf{z}}} = \text{diag} (f'_{\mathbf{z}}(z_i))_{i=1}^k, \quad D_{g_{\mathbf{z}}} = \text{diag} (g'_{\mathbf{z}}(z_i))_{i=1}^k \in \mathbb{C}^{k \times k}.$$

**Proof** This is a special case of the formula (31) described in Section 4.  $\square$

### 2.4 Simplified iteration

The simplification we propose is analogous to the idea used in the classical univariate Weierstrass iteration, which we briefly describe first. The classical univariate Weierstrass iteration finds simultaneously all roots of a given monic univariate polynomial  $f$  of degree  $k$ , and has the following simple and elegant component-wise iteration function:

$$z_i^{new} = z_i - \frac{f(z_i)}{\prod_{j \neq i} (z_i - z_j)} \quad i = 1, \dots, k.$$



One can derive this formula by applying the Newton method to the corresponding Weierstrass map, and then expressing the result in terms of the standard Lagrange polynomial basis at the iteration point: the Jacobian of the Weierstrass map is diagonal in the Lagrange basis, which results in the simple, component-wise iteration formula. Generalization of this to finding the roots of multivariate systems were proposed in [30].

Now we explore an analogue of the above simplification to our problem of solving approximate over-constrained systems. First we need to make the following assumption about the size of the support of the perturbation functions:

**Assumption:**  $|I| = |J| = k$ .

We will need the following lemma:

**Lemma 2.10** *Let  $f$ ,  $\mathbf{z}$ ,  $I$ , and  $F_I(\mathbf{z}, x)$  be as in Definition 2.6 and assume that  $|I| = k$ . Let  $L_{I,1}, \dots, L_{I,k}$  be the Lagrange polynomials defined in (5). Then for all  $1 \leq i \leq k$  we have*

$$\frac{\partial F_I(\mathbf{z}, x)}{\partial z_i} = (f'(z_i) - F'_I(\mathbf{z}, z_i)) L_{I,i}(\mathbf{z}, x). \quad (14)$$

**Proof** Implicitly differentiating the equations

$$F_I(\mathbf{z}, z_j) = f(z_j) \quad j = 1, \dots, k$$

by  $z_i$  we get

$$\frac{\partial F_I(\mathbf{z}, x)}{\partial z_i} \Big|_{x=z_j} + \delta_{i,j} \frac{\partial F_I(\mathbf{z}, x)}{\partial x} \Big|_{x=z_j} = \delta_{i,j} \frac{\partial f(x)}{\partial x} \Big|_{x=z_j}.$$

By the assumption that  $|I| = k$  we have that

$$\langle L_{I,1}, \dots, L_{I,k} \rangle = \mathbb{C}[x]_I,$$

which implies that  $\frac{\partial F_I(\mathbf{z}, x)}{\partial z_i}$  is equal to the expression in the claim.  $\square$

**Definition 2.11** *Let  $(f, g)$ ,  $k$ ,  $\mathbf{z} = (z_1, \dots, z_k)$ ,  $I$ ,  $J$ ,  $F_I(\mathbf{z}, x)$ , and  $G_J(\mathbf{z}, x)$  be as in Definition 2.6. Assume that  $|I| = |J| = k$ . As in (12), let*

$$f_{\mathbf{z}}(x) := f(x) - F_I(\mathbf{z}, x), \quad g_{\mathbf{z}}(x) := g(x) - G_J(\mathbf{z}, x).$$

*Assume that none of the  $z_i$ 's are common roots of the derivatives  $f'_{\mathbf{z}}(x)$  and  $g'_{\mathbf{z}}(x)$ . Then the **simplified Gauss-Newton iteration** with supports  $I$  and  $J$  is defined by*

$$z'_i := z_i - \frac{\overline{f'_{\mathbf{z}}(z_i)} f(z_i) + \overline{g'_{\mathbf{z}}(z_i)} g(z_i)}{|f'_{\mathbf{z}}(z_i)|^2 + |g'_{\mathbf{z}}(z_i)|^2} \quad i = 1, \dots, k. \quad (15)$$

Note that (13) equals (15) if we replace  $M_I$  and  $M_J$  by the identity matrix in (13) and exploit the diagonality of the matrices  $D_{f_{\mathbf{z}}}$  and  $D_{g_{\mathbf{z}}}$  to obtain the component-wise formulation.

The following theorem asserts that  $\mathbf{z} \in \mathbb{C}^k$  are fixed points of the simplified Gauss-Newton iteration if the corresponding perturbation functions are pointwise minimal in a neighborhood of  $\mathbf{z}$ .

**Theorem 2.12** *A point  $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{C}^k$  is a fixed point of the simplified Gauss-Newton iteration defined in (15) if there exists an open neighborhood  $U$  of  $\mathbf{z}$  such that for all  $\tilde{\mathbf{z}} = (\tilde{z}_1, \dots, \tilde{z}_k)$  and  $\mathbf{z}' = (z'_1, \dots, z'_k)$  in  $U$*

$$|F_I(\mathbf{z}, \tilde{z}_i)|^2 + |G_J(\mathbf{z}, \tilde{z}_i)|^2 \leq |F_I(\mathbf{z}', \tilde{z}_i)|^2 + |G_J(\mathbf{z}', \tilde{z}_i)|^2 \quad i = 1, \dots, k. \quad (16)$$

Note that this includes the case when  $z_1, \dots, z_k$  are common roots of  $f$  and  $g$ , in which case  $F_I(\mathbf{z}, x) = G_J(\mathbf{z}, x) = 0$ .

**Proof** Assume  $\mathbf{z} \in \mathbb{C}^k$  satisfies the condition in (16) for some neighborhood  $U$ . Then for any fixed  $\tilde{\mathbf{z}} = (\tilde{z}_1, \dots, \tilde{z}_k) \in U$  we have that for all  $i = 1, \dots, k$

$$\frac{\partial}{\partial z_i} \left( |F_I(\mathbf{z}, \tilde{z}_i)|^2 + |G_J(\mathbf{z}, \tilde{z}_i)|^2 \right) = 0.$$

Thus,

$$\frac{\partial F_I(\mathbf{z}, \tilde{z}_i)}{\partial z_i} \overline{F_I(\mathbf{z}, \tilde{z}_i)} + \frac{\partial G_J(\mathbf{z}, \tilde{z}_i)}{\partial z_i} \overline{G_J(\mathbf{z}, \tilde{z}_i)} = 0.$$

Using Lemma 2.10 we get that

$$(f'(z_i) - F'_I(\mathbf{z}, z_i)) L_{I,i}(\mathbf{z}, \tilde{z}_i) \overline{F_I(\mathbf{z}, \tilde{z}_i)} + (g'(z_i) - G'_J(\mathbf{z}, z_i)) L_{J,i}(\mathbf{z}, \tilde{z}_i) \overline{G_J(\mathbf{z}, \tilde{z}_i)} = 0.$$

In particular, as  $\tilde{\mathbf{z}}$  approaches  $\mathbf{z}$  we get that

$$(f'(z_i) - F'_I(\mathbf{z}, z_i)) \overline{f(z_i)} + (g'(z_i) - G'_J(\mathbf{z}, z_i)) \overline{g(z_i)} = 0.$$

Using the definition of  $f_{\mathbf{z}}$  and  $g_{\mathbf{z}}$  we get that  $f'_{\mathbf{z}}(z_i) \overline{f(z_i)} + g'_{\mathbf{z}}(z_i) \overline{g(z_i)} = 0$ , and the left hand side is the conjugate of the numerator of the iteration function in (15). This proves the claim.  $\square$

### 3 Multivariate Case

In this section, we describe the generalization of the results of the previous section to the multivariate setting. In the multivariate case we extend our construction to over-constrained systems of analytic functions as input, not only polynomials. Since the set of over-constrained systems of analytic functions with at least  $k$  common roots is infinite dimensional, we will restrict our objective to find the closest such system which is obtained via some perturbation from a finite dimensional “perturbation space”, given by a finite basis of analytic functions. In order to handle

analytic functions as input, we assume that they are given in a “black box” format, i.e. we assume that we can evaluate these functions in some fixed precision in unit time at any point. For our general construction we need to generalize the Lagrange interpolation to finding elements in the perturbation space with prescribed evaluations and minimal 2-norms.

**Definition 3.1** We denote by  $\mathbb{C}_n^\infty$  the set of analytic functions  $\mathbb{C}^n \rightarrow \mathbb{C}$ . Let  $\vec{f} = (f_1, \dots, f_N) \in (\mathbb{C}_n^\infty)^N$  for some  $N > n$ . For each  $i = 1, \dots, N$  let  $B_i := \{b_{i,1}, \dots, b_{i,m_i}\} \subset \mathbb{C}_n^\infty$  linearly independent over  $\mathbb{C}$ . We call  $\mathcal{P} := \bigoplus_{i=1}^N \text{span}_{\mathbb{C}}(B_i)$  the **perturbation space** with basis  $\vec{B} := (B_1, \dots, B_N)$ .

We address the following problem:

**Problem:** Given  $\vec{f} = (f_1, \dots, f_N)$  and  $\vec{B} = (B_1, \dots, B_N)$  as above. Find  $(p_1, \dots, p_N) \in \mathcal{P}$  such that  $(f_1 - p_1, \dots, f_N - p_N)$  has at least  $k$  distinct common roots in  $\mathbb{C}^n$  and  $\|p_1\|_{B_1}^2 + \dots + \|p_N\|_{B_N}^2$  is minimal. Here  $\|p_i\|_{B_i}$  denotes the 2-norm of the coefficients of  $p_i$  in the  $\mathbb{C}$ -basis  $B_i$ .

Let us define the generalized Vandermonde matrix associated with a set of basis functions  $B$ :

**Definition 3.2** Let  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_k) \in (\mathbb{C}^n)^k$ . For  $B = \{b_1, \dots, b_m\} \subset \mathbb{C}_n^\infty$  we define the **generalized Vandermonde matrix** associated with  $B$  to be the  $k \times m$  matrix with entries

$$V_B(\mathbf{z})_{i,j} := b_j(\mathbf{z}_i).$$

We denote

$$\mathcal{R}_B := \left\{ \mathbf{z} \in (\mathbb{C}^n)^k \mid \text{rank}(V_B(\mathbf{z})) = k \right\},$$

and for  $\vec{B} = (B_1, \dots, B_N)$  we use the notation  $\mathcal{R}_{\vec{B}} := \bigcap_{i=1}^N \mathcal{R}_{B_i}$ .

**Remark 3.3** We can choose the bases  $B_1, \dots, B_N$  of the perturbation space freely as long as the set  $\mathcal{R}_{\vec{B}}$  is open and everywhere dense, or it includes the possible roots we are searching for.

Now we can define the generalized multivariate Lagrange polynomials :

**Definition 3.4** Let  $B = \{b_1, \dots, b_m\} \subset \mathbb{C}_n^\infty$ . For  $\mathbf{x} \in \mathbb{C}^n$  denote  $\mathbf{x}_B = [b_1(\mathbf{x}), \dots, b_m(\mathbf{x})]$ . Let  $\mathbf{e}_1 \dots \mathbf{e}_k$  be the standard basis of  $\mathbb{C}^k$ . Let  $\mathbf{z} \in \mathcal{R}_B$ . We define the **generalized Lagrange polynomials** associated with  $B$  as  $L_{B,i}(\mathbf{z}, \mathbf{x}) := \mathbf{x}_B V_B(\mathbf{z})^\dagger \mathbf{e}_i$  for  $i = 1, \dots, k$ .

**Remark 3.5** If  $m = k$  and  $B = \{\mathbf{x}^{\alpha_1}, \dots, \mathbf{x}^{\alpha_k}\}$  for some  $\alpha_i \in \mathbb{N}^n$ , then the generalized Vandermonde matrix is a square matrix and the above formula is the one given by Ruatta in [31] for the Lagrange interpolation basis.

The following proposition is a straightforward generalization of Propositions 2.3 and 2.4.

**Proposition 3.6** *Let  $f \in \mathbb{C}_n^\infty$ ,  $B \subset \mathbb{C}_n^\infty$ ,  $|B| = m$  linearly independent over  $\mathbb{C}$ , and let  $\mathcal{P} = \text{span}_{\mathbb{C}}(B)$ . Fix  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_k) \in \mathcal{R}_B$ . Then  $L_{B,i}(\mathbf{z}, \mathbf{z}_j) = \delta_{i,j}$  for all  $i, j = 1, \dots, k$ . Furthermore, define  $p(\mathbf{z}, \mathbf{x}) = \sum_{i=1}^k f(\mathbf{z}_i) L_{B,i}(\mathbf{z}, \mathbf{x}) \in \mathcal{P}$ . Then*

$$p(\mathbf{z}, \mathbf{z}_j) = f(\mathbf{z}_j) \text{ for all } j \in \{1, \dots, k\}. \quad (17)$$

Moreover,

$$\|p\|_B^2 = \mathbf{f}^* M_B(\mathbf{z})^{-1} \mathbf{f} \quad (18)$$

is minimal among the polynomials in  $\mathcal{P}$  satisfying (17). Here

$$\mathbf{f} := (f(\mathbf{z}_1), \dots, f(\mathbf{z}_k))^T \quad \text{and} \quad M_B(\mathbf{z}) := V_B(\mathbf{z}) V_B(\mathbf{z})^* = \left( \sum_{b \in B} b(\mathbf{z}_i) \overline{b(\mathbf{z}_j)} \right)_{i,j \in \{1, \dots, k\}}. \quad (19)$$

The next theorem gives a generalization of the expressions of Karmarkar and Lakshman in [23] for the multivariate case. This is one of the main results of the paper.

**Theorem 3.7** *Let  $N > n \in \mathbb{N}$ ,  $\vec{f} = (f_1, \dots, f_N) \in (\mathbb{C}_n^\infty)^N$ ,  $\vec{B} = (B_1, \dots, B_N)$  and  $\mathcal{P} = \bigoplus_{i=1}^N \text{span}_{\mathbb{C}}(B_i)$  be as in Definition 3.1. Define  $\mathbf{f}_i(\mathbf{z}) := (f_i(\mathbf{z}_1), \dots, f_i(\mathbf{z}_k)) \in \mathbb{C}^k$  and let  $M_{B_i}(\mathbf{z})$  be as in (19) for  $i = 1, \dots, N$ . Then, if*

$$\min_{\mathbf{z} \in \mathcal{R}_{\vec{B}}} \mathbf{f}_1^* M_{B_1}^{-1} \mathbf{f}_1(\mathbf{z}) + \dots + \mathbf{f}_N^* M_{B_N}^{-1} \mathbf{f}_N(\mathbf{z}) \quad (20)$$

exists, it is equal to

$$\min_{\vec{f} \in \Omega_{\vec{B},k}(\vec{f})} \|f_1 - \tilde{f}_1\|_{B_1}^2 + \dots + \|f_N - \tilde{f}_N\|_{B_N}^2. \quad (21)$$

Here the minimum is taken within the set  $\Omega_{\vec{B},k}(f)$  defined by

$$\Omega_{\vec{B},k}(\vec{f}) := \left\{ \vec{f} = (\tilde{f}_1, \dots, \tilde{f}_N) : \forall i \ f_i - \tilde{f}_i \in \text{span}_{\mathbb{C}}(B_i), \exists (\mathbf{z}_1, \dots, \mathbf{z}_k) \in \mathcal{R}_{\vec{B}} \ \forall i, j \ \tilde{f}_i(\mathbf{z}_j) = 0 \right\}.$$

**Proof** For a fixed  $\mathbf{z} \in \mathcal{R}_{\vec{B}}$  define  $p_i(\mathbf{z}, \mathbf{x}) := \sum_{j=1}^k f_i(\mathbf{z}_j) L_{B_i,j}(\mathbf{z}, \mathbf{x}) \in \text{span}_{\mathbb{C}}(B_i)$  for all  $i = 1, \dots, N$ . Assume that the minimum in (20) exists and is taken at  $\vec{\zeta} = (\zeta_1, \dots, \zeta_k) \in \mathcal{R}_{\vec{E}}$ . Note that for all  $i \in \{1, \dots, N\}$ , if  $\tilde{f}_i$  vanishes on  $\zeta_1, \dots, \zeta_k$  and  $f_i - \tilde{f}_i \in \text{span}_{\mathbb{C}}(B_i)$ , then, by Proposition 3.6,  $\|f_i - \tilde{f}_i\|_{B_i} \geq \|p_i(\vec{\zeta}, \mathbf{x})\|_{B_i}$ . This implies that

$$(f_1(\mathbf{x}) - p_1(\vec{\zeta}, \mathbf{x}), \dots, f_N(\mathbf{x}) - p_N(\vec{\zeta}, \mathbf{x})) \in \Omega_{\vec{B},k}(\vec{f})$$

must minimize (21). The equality of (20) and (21) follows from

$$\|p_1(\vec{\zeta}, \mathbf{x})\|_{B_1}^2 + \dots + \|p_N(\vec{\zeta}, \mathbf{x})\|_{B_N}^2 = \mathbf{f}_1^* M_{B_1}^{-1} \mathbf{f}_1(\vec{\zeta}) + \dots + \mathbf{f}_N^* M_{B_N}^{-1} \mathbf{f}_N(\vec{\zeta}).$$

□

Next we define the multivariate generalization of the Weierstrass map :

**Definition 3.8** Let  $f_1, \dots, f_N \in \mathbb{C}_n^\infty$ ,  $\vec{B} = (B_1, \dots, B_N)$  and  $\mathcal{P}$  be as above. The generalized Weierstrass map is defined as follows:

$$\mathcal{W}_{\vec{B}} : \begin{cases} \mathcal{R}_{\vec{B}} & \longrightarrow \\ \vec{z} & \longmapsto \end{cases} \begin{pmatrix} \mathcal{P} \\ p_1(\vec{z}, \mathbf{x}) \\ \vdots \\ p_N(\vec{z}, \mathbf{x}) \end{pmatrix}, \quad (22)$$

where

$$p_i(\mathbf{z}, \mathbf{x}) := \sum_{j=1}^k f_i(\mathbf{z}_j) L_{B_i, j}(\mathbf{z}, \mathbf{x}) \quad i = 1, \dots, N.$$

The next proposition is a straightforward generalization of Proposition 2.7 :

**Proposition 3.9** Let  $\vec{f} = (f_1, \dots, f_N) \in (\mathbb{C}_n^\infty)^N$ ,  $\vec{B} = (B_1, \dots, B_N)$  be as above. Then for all  $\vec{z} \in \mathcal{R}_{\vec{B}}$  we have  $\mathcal{W}_{\vec{B}}(\vec{z}) = 0$  if and only if  $\{\mathbf{z}_1, \dots, \mathbf{z}_k\}$  are common roots of  $f_1, \dots, f_N$ . Moreover, using the notation of Theorem 3.7, we have

$$\min_{\vec{z} \in \mathcal{R}_{\vec{B}}} \|\mathcal{W}_{\vec{B}}(\vec{z})\|^2 = \min_{\vec{f} \in \Omega_{\vec{B}, k}(\vec{f})} \|f_1 - \tilde{f}_1\|_{B_1}^2 + \dots + \|f_N - \tilde{f}_N\|_{B_N}^2. \quad (23)$$

In the rest of the paper we will describe iterative methods to approximate the minimum

$$\min_{\vec{z} \in \mathcal{R}_{\vec{B}}} \|\mathcal{W}_{\vec{B}}(\vec{z})\|^2.$$

## 4 Numerical methods

In this section we describe the iterative methods we use in our numerical experiments for comparison. These methods try to minimize the squared 2-norm of a function  $W : U \rightarrow \mathbb{C}^T$  for some open subset  $U \subseteq \mathbb{C}^S$ , by approximating it by its truncated Taylor series expansion.

### 4.1 Gauss-Newton method

Using the previous notation, in our case  $W := \mathcal{W}_{\vec{B}} : \mathcal{R}_{\vec{B}} \rightarrow \mathcal{P}$  is the generalized Weierstrass map defined in Definition 3.8, such that its image is expressed as the vector of coefficients of the perturbation polynomials in  $\mathcal{P}$ . We denote by  $\nabla$  the vector of derivations by the variables  $\vec{z}$  (and not by their conjugates), and  $J = \nabla W$ . We also denote by  $\overline{\nabla}$  the vector of derivations by the conjugate variables. To minimize indices and simplify the notation, we denote by  $z_i$  and  $\bar{z}_i$  the coordinates of  $\vec{z}$  and their conjugates.

First we argue that it is sufficient to consider only derivations by the variables  $\vec{z}$  and not by their conjugates when we define the Gauss-Newton method. We need the following lemmas:

**Lemma 4.1** *Let  $\vec{f} = (f_1, \dots, f_N)$ ,  $\vec{B} = (B_1, \dots, B_N)$ ,  $\mathbf{z} \in \mathcal{R}_{\vec{B}} \subset (\mathbb{C}^n)^k$  as in Definition 3.8. Define  $F$  to be the column vector*

$$F := (f_1(\mathbf{z}_1), \dots, f_1(\mathbf{z}_k), \dots, f_N(\mathbf{z}_1), \dots, f_N(\mathbf{z}_k))^T \in \mathbb{C}^{kN}$$

*and the matrix  $V$  as the block diagonal matrix of size  $(kN) \times (\sum |B_i|)$ , with diagonal blocks the Vandermonde matrices  $V_{B_1}(\mathbf{z}), \dots, V_{B_N}(\mathbf{z})$ . Then the gradient of the Weirstrass map  $W$  is*

$$J = \nabla W = V^\dagger (\nabla F - (\nabla V)W). \quad (24)$$

**Proof** By definition,  $W$  is the least square solution of

$$F = VW.$$

The use of the Moore-Penrose pseudoinverse of  $V$  can be described in two steps. First we find  $G$  such that

$$VV^*G = F \quad (25)$$

then we compute  $W$  as

$$W = V^*G. \quad (26)$$

From equation (26) we have

$$\nabla W = (\nabla V^*)G + V^*(\nabla G). \quad (27)$$

From equation (25) we have

$$\nabla G = (VV^*)^{-1} (\nabla F - (\nabla V)V^*G - V(\nabla V^*)G). \quad (28)$$

Combining equations (27) and (28) and using that fact that  $\nabla V^* = 0$  gives

$$\nabla W = V^*(VV^*)^{-1} (\nabla F - (\nabla V)V^*(VV^*)^{-1}F) = V^\dagger (\nabla F - (\nabla V)W).$$

□

**Lemma 4.2**

$$\frac{\partial W^*}{\partial z_i} V^\dagger = 0.$$

**Proof** By definition  $W = V^\dagger F = V^*(VV^*)^{-1}F$  and thus

$$\frac{\partial W^*}{\partial z_i} = -F^*(VV^*)^{-1} \frac{\partial V}{\partial z_i} V^*(VV^*)^{-1}V + F^*(VV^*)^{-1} \frac{\partial V}{\partial z_i}.$$

Then we have

$$\begin{aligned} \frac{\partial W^*}{\partial z_i} V^\dagger &= F^*(VV^*)^{-1} \left( \frac{\partial V}{\partial z_i} - \frac{\partial V}{\partial z_i} V^*(V^\dagger)^* \right) V^\dagger \\ &= F^*(VV^*)^{-1} \left( \frac{\partial V}{\partial z_i} V^\dagger - \frac{\partial V}{\partial z_i} V^*(V^\dagger)^* V^\dagger \right) \\ &= F^*(VV^*)^{-1} \left( \frac{\partial V}{\partial z_i} V^\dagger - \frac{\partial V}{\partial z_i} V^*(VV^*)^{-1} V V^*(VV^*)^{-1} \right) \\ &= F^*(VV^*)^{-1} \left( \frac{\partial V}{\partial z_i} V^\dagger - \frac{\partial V}{\partial z_i} V^\dagger \right) = 0 \quad \blacksquare \end{aligned}$$

□

**Corollary 4.3**

$$\frac{\partial W^*}{\partial z_i} W = 0, \quad \frac{\partial W^*}{\partial z_i} J = 0, \quad \text{and} \quad \frac{\partial \|W\|^2}{\partial z_i} = W^* \frac{\partial W}{\partial z_i}$$

**Proof** Follows from  $W = V^\dagger F$  and  $J = V^\dagger(\nabla F - (\nabla V)W)$ . □

**Corollary 4.4** *If we assume that  $J^*(\xi)W(\xi) = 0$  then*

$$\frac{\partial J^\dagger W}{\partial \bar{z}_i}(\xi) = \left( (J^* J)^{-1} \left( \frac{\partial J}{\partial z_i} \right)^* W \right)(\xi).$$

**Proof** Using that  $J^*(\xi)W(\xi) = 0$  we get

$$\begin{aligned} \frac{\partial J^\dagger W}{\partial \bar{z}_i}(\xi) &= \frac{\partial (J^* J)^{-1} J^* W}{\partial \bar{z}_i}(\xi) \\ &= \left( (J^* J)^{-1} \frac{\partial J^*}{\partial \bar{z}_i} W \right)(\xi) + \left( (J^* J)^{-1} J^* \frac{\partial W}{\partial \bar{z}_i} \right)(\xi) \\ &= \left( (J^* J)^{-1} \left( \frac{\partial J}{\partial z_i} \right)^* W \right)(\xi) + \left( (J^* J)^{-1} \left( \frac{\partial W^*}{\partial z_i} J \right)^* \right)(\xi) \end{aligned}$$

and the last term is 0 by the previous corollary. □

The following argument is from [8, Theorem 4]:

**Proposition 4.5** *Define the Gauss-Newton method by the map*

$$N_W(\vec{z}) := \vec{z} - J^\dagger(\vec{z})W(\vec{z}). \quad (29)$$

Let  $\xi \in \mathcal{R}_{\bar{B}}$  such that  $J$  has full rank at  $\xi$ ,

$$J^*(\xi)W(\xi) = 0,$$

and we have the following inequality:

$$\|J^\dagger(\xi)\|^2 \cdot \left\| \begin{bmatrix} \nabla J(\xi) \\ \nabla J^*(\xi) \end{bmatrix} \right\| \cdot \|W(\xi)\| < 1, \quad (30)$$

where for a matrix  $M$ ,  $\|M\|$  denotes the operator 2-norm, i.e.  $\|M\| = \sup_{\|x\|=1} \|Mx\|$ , while for a 3-dimensional matrix  $N$  it is  $\|N\| = \sup_{\|x\|=1} \|N(x, x)\|$ . Then  $\xi$  is an attractive fixed point for  $N_W$ .

**Proof** To prove the claim we have that

$$N_W(\xi, \bar{\xi}) - N_W(\mathbf{z}, \bar{\mathbf{z}}) = \begin{bmatrix} \nabla N_W(\xi) & \bar{\nabla} N_W(\xi) \end{bmatrix} \cdot \begin{bmatrix} \xi - \mathbf{z} \\ \bar{\xi} - \bar{\mathbf{z}} \end{bmatrix} + h.o.t.$$

Using that  $J^*(\xi)W(\xi) = 0$  we get that

$$\nabla N_W(\xi) = -(J^*J)^{-1}(\nabla J^*)W(\xi),$$

and also using the previous Corollary we have that

$$\bar{\nabla} N_W(\xi) = -(J^*J)^{-1}(\nabla J)^*W(\xi).$$

Therefore,

$$\begin{bmatrix} \nabla N_W(\xi) & \bar{\nabla} N_W(\xi) \end{bmatrix} = -(J^*J)^{-1}(\xi) \begin{bmatrix} \nabla J(\xi) & \nabla J^*(\xi) \end{bmatrix} W(\xi),$$

and its norm is bounded by  $\|J^\dagger(\xi)\|^2 \cdot \|\begin{bmatrix} \nabla J(\xi) & \nabla J^*(\xi) \end{bmatrix}\| \cdot \|W(\xi)\| < 1$ , which proves that  $\xi$  is an attractive fixed point of  $N_W$ .  $\square$

Next we give an explicit formula for the Gauss-Newton iteration defined in (29) in terms of  $M_{B_i}(\bar{\mathbf{z}}) = V_{B_i}(\bar{\mathbf{z}})V_{B_i}^*(\bar{\mathbf{z}})$  and the function values  $\mathbf{f}_i(\bar{\mathbf{z}})$ .

**Proposition 4.6** *Using the notation of Theorem 3.7, the iteration defined by*

$$\mathbf{z}' = \mathbf{z} - \left( \sum_{i=1}^N D_i^* M_{B_i}^{-1} D_i \right)^{-1} \left( \sum_{i=1}^N D_i^* M_{B_i}^{-1} \mathbf{f}_i \right). \quad (31)$$

*is the Gauss-Newton iteration defined in (29) for the Weierstrass map  $\mathcal{W}_{\bar{B}}$ . Here for  $i = 1, \dots, N$*

$$\mathbf{f}_i := (f_i(\mathbf{z}_1), \dots, f_i(\mathbf{z}_k))^T, \quad M_{B_i} = V_{B_i} V_{B_i}^* \text{ and } D_i := \begin{bmatrix} D_{i,1} & & & \\ & D_{i,2} & & \\ & & \ddots & \\ & & & D_{i,k} \end{bmatrix} \in \mathbb{C}^{k \times nk}$$

*with each block  $D_{i,j}$  of size  $1 \times n$  and defined as  $D_{i,j} := \left[ \frac{\partial(f_i - p_i)}{\partial x_s}(\mathbf{z}_j) \right]_{1 \leq s \leq n}$ .*

**Proof**  $N_W$  in (29) uses the pseudo-inverse  $J^\dagger$ . We can expand the pseudo-inverse of  $(V^\dagger(\nabla F - \nabla VW))$  as follows:

$$\begin{aligned} & \left( V^\dagger(\nabla F - \nabla VW) \right)^\dagger \\ &= \left( \left( V^\dagger(\nabla F - \nabla VW) \right)^* \left( V^\dagger(\nabla F - \nabla VW) \right) \right)^{-1} \left( V^\dagger(\nabla F - \nabla VW) \right)^* \\ &= ((\nabla F - \nabla VW)^* (VV^*)^{-1} (\nabla F - \nabla VW))^{-1} (\nabla F - \nabla VW)^* (V^\dagger)^* \end{aligned}$$



using the fact that

$$(V^\dagger)^* V^\dagger = \left( (VV^*)^{-1} \right)^* VV^* (VV^*)^{-1} = \left( (VV^*)^{-1} \right)^* = (VV^*)^{-1}.$$

When this is substituted into (29) we get

$$\mathbf{z}' = \mathbf{z} - \left( ((\nabla F - \nabla VW)^* (VV^*)^{-1} (\nabla F - \nabla VW))^{-1} (\nabla F - \nabla VW)^* (VV^*)^{-1} F \right) (\mathbf{z}) \quad (32)$$

To get (31) from (32) we observe that  $(\nabla V)$  is a 3-dimensional matrix of size  $(kN) \times \left( \sum_{t=1}^N |B_t| \right) \times (kn)$  consisting of the  $kn$  block diagonal matrices  $\frac{\partial V}{\partial z_{i,j}}$  for  $i = 1, \dots, k, j = 1, \dots, n$ . In each block of  $\frac{\partial V}{\partial z_{i,j}}$  only one row is non-zero, the one corresponding to  $\mathbf{z}_i$ , and the entries of this row are changed from  $b(\mathbf{z}_i)$  to  $\frac{\partial b}{\partial x_j}(\mathbf{z}_i)$  for  $b$  in some  $B_t$ . Since  $W$  is the vector consisting of the coefficient vectors of  $p_1, \dots, p_N$  in the bases  $B_1, \dots, B_N$ , we conclude that  $\nabla F - (\nabla V)W$  is a  $(kN) \times (kn)$  matrix with columns corresponding to the partial derivatives  $\frac{\partial}{\partial z_{i,j}}$  ( $i = 1, \dots, k, j = 1, \dots, n$ ), and each of these columns have 0 entries everywhere except in the  $i + (t-1)k$ -th place for  $t = 1, \dots, N$ , where they are equal to  $\frac{\partial(f_t - p_t)}{\partial x_j}(\mathbf{z}_i)$ . To get (31), we use the block diagonal structure of  $(VV^*)^{-1}$  with blocks  $M_{B_t}^{-1}$  ( $t = 1, \dots, N$ ). □

## 4.2 Simplified Gauss-Newton method

In this section we describe the generalization of the univariate simplified Gauss-Newton iteration defined in Definition 2.11. First we show how the simplified Gauss-Newton method is obtained from the standard Gauss-Newton method by making some adjustments based on the specifics of this particular minimization problem. Although this method does not find a minimum in the 2-norm, as we shall see, it does find a minimum that is reasonable in the context of the problem while using significantly reduced computational effort.

Consider the formula we obtained in (32) for the Gauss-Newton iteration. What we want is to find a way to simplify this formula to a form that can be more efficiently computed. If  $V$  were a square unitary matrix, the  $(VV^*)$  terms would be the identity matrix and would disappear from the formula.  $V$  is unlikely to be unitary, but it turns out that if we perform this cancellation anyway, we get a new formula that can be computed more efficiently than that of the standard Gauss-Newton, and surprisingly we still converge to a set of polynomials that can be said to be locally minimally distant from the originals—if we use a different method for measuring distance. Dropping  $(VV^*)^{-1}$  we get

$$\mathbf{z}' = \mathbf{z} - \left( ((\nabla F - (\nabla V)W)^* (\nabla F - (\nabla V)W))^{-1} (\nabla F - (\nabla V)W)^* \right) (\mathbf{z}) F(\mathbf{z}),$$

which reduces to the *simplified Gauss-Newton iteration* formula

$$\mathbf{z}' = \mathbf{z} - (\nabla F - (\nabla V)W)^\dagger (\mathbf{z}) F(\mathbf{z}). \quad (33)$$

In order to turn (33) into a component-wise iteration function, as in the univariate case, we need the following assumption:

$$\textbf{Assumption : } |B_1| = \dots = |B_N| = k. \quad (34)$$

Then we can prove the following generalization of Lemma 2.10, implying the simple structure of the partial derivatives of the Weierstrass map, when expressed in terms of the Lagrange basis:

**Lemma 4.7** *Let  $f \in \mathbb{C}_n^\infty$ ,  $B \subset \mathbb{C}_n^\infty$ , and assume that  $|B| = k$ . For a fixed  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_k) \in \mathcal{R}_B$  let the Lagrange polynomials  $L_{B,i}(\mathbf{z}, \mathbf{x})$  ( $i = 1, \dots, k$ ) defined as in Definition 3.4, and as before, let*

$$p(\mathbf{z}, \mathbf{x}) := \sum_{i=1}^k f(\mathbf{z}_i) L_{B,i}(\mathbf{z}, \mathbf{x}).$$

Then

$$\frac{\partial p}{\partial z_{i,j}}(\mathbf{z}, \mathbf{x}) = \left( \frac{\partial(f-p)}{\partial x_j}(\mathbf{z}_i) \right) L_{B,i}(\mathbf{z}, \mathbf{x}).$$

**Proof** The proof is similar to the proof of Lemma 2.10, and it is based on computing the evaluations of  $\frac{\partial p}{\partial z_{i,j}}$  at  $\mathbf{x} = \mathbf{z}_t$  for  $t = 1, \dots, k$ . Then from  $|B| = k$  and  $\mathbf{z} \in \mathcal{R}_B$  it follows that  $\{L_{B,1}, \dots, L_{B,k}\}$  generates  $\text{span}_{\mathbb{C}} B$ , thus these evaluations uniquely determine the elements  $\text{span}_{\mathbb{C}} B$ .  $\square$

Using the previous lemma we can give the following simple component-wise formula for the simplified Gauss-Newton iteration:

**Definition 4.8** *Let  $\vec{f} = (f_1, \dots, f_N)$  and  $\vec{B} = (B_1, \dots, B_N)$  be as above. Let  $(p_1(\mathbf{z}, \mathbf{x}), \dots, p_N(\mathbf{z}, \mathbf{x})) \in \mathcal{P}$  be as in Theorem 3.7. Fix  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_k) \in \mathcal{R}_{\vec{B}}$ . Assume that  $|B_i| = k$  for all  $i = 1, \dots, N$ . Define*

$$\vec{f}_{\mathbf{z}}(\mathbf{x}) := (f_1(\mathbf{x}) - p_1(\mathbf{z}, \mathbf{x}), \dots, f_N(\mathbf{x}) - p_N(\mathbf{z}, \mathbf{x})).$$

*Let  $J_{\mathbf{z}}(\mathbf{x})$  be the  $N \times n$  Jacobian matrix of  $\vec{f}_{\mathbf{z}}(\mathbf{x})$ . Assume that  $\text{rank}(J_{\mathbf{z}}(\mathbf{z}_i)) = n$  for all  $i = 1, \dots, k$ . Then the **simplified Gauss-Newton iteration** is defined by*

$$\mathbf{z}'_i := \mathbf{z}_i - J_{\mathbf{z}}(\mathbf{z}_i)^\dagger \vec{f}(\mathbf{z}_i) \quad i = 1, \dots, k. \quad (35)$$

The following theorem is a generalization of Theorem 2.12 and asserts that  $\mathbf{z} \in (\mathbb{C}^n)^k$  is a fixed point of the simplified Gauss-Newton iteration if it corresponds to perturbation functions which are locally pointwise minimal.

**Theorem 4.9** *A point  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_k) \in \mathcal{R}_{\vec{B}}$  is a fixed point of the simplified Gauss-Newton iteration in (35) if there exists an open neighborhood  $U$  of  $\mathbf{z}$  such that for all  $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_k)$  and  $\mathbf{z}' = (\mathbf{z}'_1, \dots, \mathbf{z}'_k)$  in  $U$  and for all  $i = 1, \dots, k$*

$$|p_1(\mathbf{z}, \tilde{\mathbf{z}}_i)|^2 + \dots + |p_1(\mathbf{z}, \tilde{\mathbf{z}}_i)|^2 \leq |p_1(\mathbf{z}', \tilde{\mathbf{z}}_i)|^2 + \dots + |p_1(\mathbf{z}', \tilde{\mathbf{z}}_i)|^2. \quad (36)$$

*Note that this includes the case when  $\mathbf{z}_1, \dots, \mathbf{z}_k$  are common roots of  $f_1, \dots, f_N$ , in which case  $p_1(\mathbf{z}, \mathbf{x}) = \dots = p_N(\mathbf{z}, \mathbf{x}) = 0$ .*

**Proof** For  $\mathbf{z}$  to be a fixed point for the simplified Gauss-Newton iteration, it is sufficient to prove that for all  $i = 1, \dots, k$   $J_{\mathbf{z}}(\mathbf{z}_i)^* \vec{f}(\mathbf{z}_i) = 0$ , which is equivalent to

$$\sum_{t=1}^N \frac{\partial(f_t - p_t)}{\partial x_j}(\mathbf{z}_i) \overline{f_t(\mathbf{z}_i)} = 0 \text{ for all } i = 1, \dots, k, j = 1, \dots, n. \quad (37)$$

By (36) we have that for any  $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_k) \in U$  and for all  $i = 1, \dots, k$  and  $j = 1, \dots, n$

$$\frac{\partial}{\partial z_{i,j}} (|p_1(\mathbf{z}, \tilde{\mathbf{z}}_i)|^2 + \dots + |p_1(\mathbf{z}, \tilde{\mathbf{z}}_i)|^2) = 0,$$

which implies that

$$\sum_{t=1}^N \overline{p_t(\mathbf{z}, \tilde{\mathbf{z}}_i)} \frac{\partial p_t(\mathbf{z}, \tilde{\mathbf{z}}_i)}{\partial z_{i,j}} = 0.$$

Using Lemma 4.7 we get that

$$\sum_{t=1}^N \overline{p_t(\mathbf{z}, \tilde{\mathbf{z}}_i)} \left( \frac{\partial(f_t - p_t)}{\partial x_j}(\mathbf{z}_i) \right) L_{B,i}(\mathbf{z}, \tilde{\mathbf{z}}_i) = 0.$$

Approaching with  $\tilde{\mathbf{z}}$  to  $\mathbf{z}$  we get (37). □

### 4.3 Quadratic Iteration

The quadratic iteration method explicitly calculates the Gradient and Hessian of the function  $W^*W$ , evaluates them at the current point  $\mathbf{z}$ , and directly solves for the critical point  $\mathbf{z}$  using the linear system

$$0 = G(\mathbf{z}_0, \bar{\mathbf{z}}_0) + H(\mathbf{z}_0, \bar{\mathbf{z}}_0) \begin{bmatrix} \mathbf{z} - \mathbf{z}_0 \\ \bar{\mathbf{z}} - \bar{\mathbf{z}}_0 \end{bmatrix}. \quad (38)$$

Since such a calculated critical point is as likely to be a maximum (or saddle) point as a minimum, usage of  $H$  is adjusted by removing positive eigenvalues to ensure movement towards a desired minimum. Additionally, if the 2-norm of  $W$  at  $\mathbf{z}'$  is greater than that at  $\mathbf{z}$ , points along the line between  $\mathbf{z}'$  and  $\mathbf{z}$  closer and closer to  $\mathbf{z}$  are tested until a decrease in the norm is detected.

### 4.4 Conjugate Gradient method

The conjugate gradient method does repeated one dimensional minimizations, in a single direction for each iteration, until a local minimum in all directions is found. We will label the directions used for iteration  $i$  as  $g_i$ . These directions are not chosen randomly, but in such a way as to find the minimum in as few iterations as possible. Below we will show that for quadratic functions the minimum will be found in at most  $n$  iterations.

A general quadratic function in  $n$  variables has the form

$$Q(\mathbf{z}) = K + \mathbf{z}^T L + \mathbf{z}^T M \mathbf{z}$$

where  $K$  is a scalar,  $L$  is an  $n$  dimensional vector, and  $M$  is an  $n \times n$  matrix. This method assumes that the function is real-valued and that the quadratic term,  $\mathbf{z}^T M \mathbf{z}$ , is nonnegative for all  $\mathbf{z}$ . Otherwise it does not make sense to talk of the function's minimum.

To simplify the discussion, we will work with a translated version of this quadratic function,  $Q(\mathbf{z} + \mathbf{z}_0) - Q(\mathbf{z}_0)$ , so that the starting point of the iteration is at the origin, and the value of the function at the origin is zero. We can then assume that our quadratic function  $Q$  has the form

$$Q = \mathbf{z}^T L + \mathbf{z}^T M \mathbf{z}.$$

The key idea behind the conjugate gradient method is to choose each iteration's search direction  $g_i$  to be *conjugate* to the previous directions, which means that

$$g_i^T M g_j = 0 \quad \forall j < i.$$

Then for any linear combination of these conjugate directions,  $\sum_{j=1}^n a_j g_j$  with  $a_j \in \mathbb{C}$ , we have

$$\begin{aligned} Q \left( \sum_{j=1}^n a_j g_j \right) &= \sum_{j=1}^n (a_j g_j^T) L + \sum_{j=1}^n (a_j g_j^T) M \sum_{j=1}^n (a_j g_j) \\ &= \sum_{j=1}^n (a_j g_j^T L + a_j g_j^T M a_j g_j) \\ &= \sum_{j=1}^n Q(a_j g_j). \end{aligned}$$

So minimization of  $Q$  can occur independently in each of the conjugate directions. It must be complete after  $n$  iterations since all possible search directions will have been exhausted.

By using calculated gradient information, the optimization directions are each chosen to be as close to the direction of steepest descent of the function as possible while maintaining the required conjugacy relationship. This allows the method to stop in fewer iterations when there are some directions that are already at or near a minimum.

When minimizing functions that are not precisely quadratic, such as the problem we are dealing with, the exact solution is not guaranteed to be found within  $n$  iterations since the effects of the  $g_i$  vectors on the value of the function are not independent. However, the practice of following the steepest conjugate directions first can still allow us to come acceptably close to the solution within  $n$  iterations depending on the characteristics of our function and our required tolerance.

It should be noted that conjugate directions can be calculated without using the matrix  $M$ . This saves significant computation time by avoiding calculation of the Hessian which would otherwise be required when using a quadratic Taylor series approximation.

## 5 Algorithmic Complexity

The per iteration operation counts are represented in the following table, where

$N$  is the number of input (and output) functions;

$n$  is the number of variables used in the input functions;

$k$  is the number of input (and output) roots;

$\beta$  is the number of bits of accuracy used for the intermediate steps of the conjugate gradient method

Here we make the assumption that all perturbation bases  $B_1, \dots, B_N$  has cardinality  $k$ .

Method	Input Evaluations	Basis Evaluations	Arithmetic Operations
Simp G-N	$\mathcal{O}(N \cdot k \cdot n)$	$\mathcal{O}(N \cdot k^2 \cdot n)$	$\mathcal{O}(\max(N \cdot k^3, N \cdot k \cdot n^2))$
Std G-N	$\mathcal{O}(N \cdot k \cdot n)$	$\mathcal{O}(N \cdot k^2 \cdot n)$	$\mathcal{O}(N \cdot k^3 \cdot n^2)$
Quad It	$\mathcal{O}(N \cdot k \cdot n^2)$	$\mathcal{O}(N \cdot k^2 \cdot n^2)$	$\mathcal{O}(N \cdot k^3 \cdot n^2)$
Conj Grd	$\mathcal{O}(N \cdot k \cdot (n + \beta))$	$\mathcal{O}(N \cdot k^2 \cdot (n + \beta))$	$\mathcal{O}(N \cdot k^3 \cdot (n + \beta))$

The *Input Evaluations* column is the number of evaluations of input functions or their derivatives. The *Basis Evaluations* column is the number of evaluations of perturbation basis functions or their derivatives. The *Arithmetic Operations* column is the number of simple scalar arithmetic operations, excluding the operations involved in evaluating the functions from the preceding two columns.

Calculation of the gradient of our 2-norm requires evaluation of  $n$  partial derivatives at each of  $k$  input roots for each of the  $N$  input functions and  $N$  perturbation functions, for a total of  $N \cdot k \cdot n$  evaluations and, since each perturbation functions are the sum of  $k$  basis functions,  $N \cdot k^2 \cdot n$  basis evaluations. This accounts for the  $N \cdot k \cdot n$  input evaluations and  $N \cdot k^2 \cdot n$  basis evaluations for the two Gauss-Newton methods and the conjugate gradient method.

The number of function evaluations for the quadratic iteration method is dominated by the calculation of the Hessian matrix, which the other methods avoid. The Hessian requires evaluation at  $n^2$  partial derivatives for each of  $N$  input functions and  $N$  perturbation functions at  $k$  different points. This is a factor of  $n$  more evaluations than is required by the gradient calculation, giving us  $N \cdot k \cdot n^2$  input evaluations and  $N \cdot k^2 \cdot n^2$  basis function evaluations.

Each method starts by calculating the basis function coefficients for the perturbation function at the current iteration point. This requires the solution of  $N$  different linear systems. Since the Vandermonde matrices have dimension  $k \times k$ , each of this steps requires  $\mathcal{O}(k^3)$  operations.

Furthermore, the Simplified Gauss-Newton method requires  $\mathcal{O}(N \cdot n^2)$  operations to solve each of the  $k$  equations in formula (35). This requires effort  $\mathcal{O}(N \cdot k \cdot n^2)$ . This may be greater or less than the effort to solve the above Vandermonde system, so the complexity is determined to be the greater of  $\mathcal{O}(N \cdot k^3)$  and  $\mathcal{O}(N \cdot k \cdot n^2)$ . If the solution of the Vandermonde system is

the dominating factor, further savings can be realized if all of the input functions use the same perturbation basis. The complexity is then the greater of  $\mathcal{O}(k^3)$  and  $\mathcal{O}(N \cdot k \cdot n^2)$ .

The standard Gauss-Newton method requires  $\mathcal{O}(k^3 n^3)$  operations to solve equation (31) since the matrix to be inverted is a  $nk \times nk$  matrix, plus  $\mathcal{O}(Nk^2 n^2)$  additions to compute the sum of  $N$  matrices each of size  $nk \times nk$ . These can be bounded by  $\mathcal{O}(Nk^3 n^2)$  since  $N > n$ .

The quadratic iteration method requires  $N \cdot k$  operations to calculate each entry of the  $nk \times nk$  Hessian matrix. This is because each of  $N$  perturbation functions contributes to every matrix entry and there are  $k$  basis function evaluations that need to be combined to get each perturbation function evaluation. Solution of the linear system (38) involving this matrix requires  $\mathcal{O}(k^3 \cdot n^3)$  operations. Since  $N > n$ , it is the setup of the Hessian that dominates, which requires  $\mathcal{O}(N \cdot k^3 \cdot n^2)$  operations.

The  $\beta$  factor for the conjugate gradient method comes from the line minimization performed during each step. The method assumes that the directional derivative along the line is zero at the minimum. Thus the more accurate the minimization, the more accurate this assumption. The factor of  $\beta$  is the average number of steps to arrive at this minimization to machine precision. Some functions' line minimums are found more rapidly than this and for some functions less precision can be used without sacrificing convergence rate.

In most cases the simplified Gauss-Newton method does the fewest operations per iteration by a factor of  $k$ . For some problems (i.e. where  $n^2 > k^2 \cdot (n + \beta)$ ) the conjugate gradient method appears that it would provide better performance. Tests indicate that for problems this complicated the conjugate gradient method is unlikely to converge to a good local minimum (i.e. a minimum close to the global minimum), so using the simplified Gauss-Newton would still be the recommended method. Although the quadratic iteration and standard Gauss-Newton methods have the same reported number of operations per iteration, quadratic iteration is actually a nontrivial constant factor slower than the standard Gauss -Newton method.

## 6 Comparison Tests

### 6.1 Test Design

Tests were performed using four different configuration. The configurations differed in the numbers of polynomials ( $N$ ), variables ( $n$ ), degrees ( $D$ ), and number of common roots ( $k$ ) for which to search.

Each random polynomial was generated by creating all monomials of total degree less than or equal to  $D$ , the degree chosen for that problem, then applying a randomly generated coefficient between  $-100$  and  $100$ .  $k$  random points were then chosen in the range  $(-10, 10)$ . Polynomials were then generated that interpolated each of these random polynomials at each of the random

points. These interpolating polynomials were subtracted from the original random polynomials to give a system with  $k$  common roots that are referred to as the unperturbed polynomials.

A perturbation basis ( $B$ ) was chosen using  $k$  monomials of smallest total degree. The input polynomials were generated from these unperturbed polynomials by adding to each polynomial a randomly generated polynomial with terms chosen from the perturbation basis. Each of these randomly generated polynomials is created as  $\sum_{i=1}^k r_i \cdot B_i$ , where each  $r_i$  is a different randomly generated number and  $B_i$  is the  $i$ th element of the perturbation basis  $B$ . For each set of tests,  $r_i$  was chosen in the five different ranges  $(-10^x, 10^x)$  for  $x \in \{-2, -1, 0, 1, 2\}$ . Ten problems were run for each range, making a total of fifty problems per configuration. The starting point for each iteration was chosen as the roots of the unperturbed polynomials, modified by adding a vector randomly chosen within the unit hypersphere.

## 6.2 Tables

Method	%Con- verged	Rel Residual			Abs Resid		Rel Output Norm			Abs Output Norm			Iter Cnt
		Min	Avg	Max	Min	Max	Min	Avg	Max	Min	Avg	Max	
Simp G-N	100	1.00	1.00	1.00	4.7e-7	0.80	1.00	1.00	1.00	2.9e-4	0.04	0.34	4.42
Std G-N	100	1.00	1.00	1.00	4.7e-7	0.80	1.00	1.00	1.00	2.9e-4	0.04	0.34	4.42
Quad It	100	1.00	1.00	1.00	4.7e-7	0.80	1.00	1.00	1.00	2.9e-4	0.04	0.34	4.98
Conj Grd	100	1.00	1.00	1.00	4.7e-7	0.80	1.00	1.00	1.00	2.9e-4	0.04	0.34	4.20

5 polynomials of degree 3 in 1 variable with 1 common root.

There were 50 problems for which all methods converged.

Method	%Con- verged	Rel Residual			Abs Resid		Rel Output Norm			Abs Output Norm			Iter Cnt
		Min	Avg	Max	Min	Max	Min	Avg	Max	Min	Avg	Max	
Simp G-N	98	1.00	1.00	1.00	1.3e-5	0.85	1.00	1.00	1.00	9.5e-4	0.05	0.28	4.67
Std G-N	100	1.00	1.16	1.45	1.4e-5	0.97	0.64	0.88	1.02	8.1e-4	0.04	0.22	4.86
Quad It	100	1.00	1.16	1.45	1.4e-5	0.97	0.64	0.88	1.02	8.1e-4	0.04	0.22	8.08
Conj Grd	100	1.04	957	1.2e4	0.04	1.02	0.70	14.5	87.4	0.03	0.09	0.22	25.96

5 polynomials of degree 2 in 2 variables with 2 common roots.

There were 49 problems for which all methods converged.

Method	%Con- verged	Rel Residual			Abs Resid		Rel Output Norm			Abs Output Norm			Iter Cnt
		Min	Avg	Max	Min	Max	Min	Avg	Max	Min	Avg	Max	
Simp G-N	70	1.00	1.00	1.00	1.9e-5	0.31	1.00	1.00	1.00	2.5e-3	0.05	0.24	6.69
Std G-N	76	1.14	1.62	2.64	2.4e-5	0.67	0.30	0.40	0.51	7.7e-4	0.02	0.12	6.31
Quad It	92	1.14	15.0	427	2.4e-5	0.69	0.30	0.69	8.31	7.7e-4	0.02	0.11	29.29
Conj Grd	90	2.65	3.5e3	4.6e4	0.30	0.91	0.69	21.6	94.0	0.13	0.18	0.25	19.97

5 polynomials of degree 2 in 4 variables with 6 common roots.

There were 35 problems for which all methods converged.

Method	%Con- verged	Rel Residual			Abs Resid		Rel Output Norm			Abs Output Norm			Iter Cnt
		Min	Avg	Max	Min	Max	Min	Avg	Max	Min	Avg	Max	
Simp G-N	94	1.00	1.00	1.00	3.1e-5	0.73	1.00	1.00	1.00	1.9e-3	0.06	0.33	8.05
Std G-N	98	0.95	1.18	1.46	3.6e-5	0.87	0.39	0.54	0.72	1.0e-3	0.03	0.16	5.79
Quad It	100	0.95	2.17	42.8	3.6e-5	0.87	0.39	0.61	3.70	1.0e-3	0.03	0.16	20.38
Conj Grd	90	1.07	2.3e3	2.7e4	0.25	0.95	0.57	24.5	118	0.11	0.20	0.31	22.93

9 polynomials of degree 2 in 4 variables with 6 common roots.

There were 42 problems for which all methods converged.

## 6.3 Explanation of Tables

The first column of the tables names the method used in the test. *Simp G-N* is the simplified Gauss-Newton, *Std G-N* is the standard Gauss-Newton method, *Quad It* is the quadratic iteration method, and *Conj Grad* is the conjugate gradient method.

All calculated values except the convergence percentage are measuring only the results from the problems for which all methods converged. This way we ensure that the numbers from each



method are comparable.

The *Converge %* column indicates the percentage of problems for which the method converged. For these tests, a method is said to have converged if within 128 iterations the change produced during each of two consecutive iterations is less than 0.001. For the Gauss-Newton type methods, if three consecutive iterations have increasing step size, the method is considered to be diverging. The quadratic iteration and conjugate gradient methods are designed such that each step guaranteed to move closer to the desired local minimum so no divergence test is done.

The following three columns report a relative residual, where residual is the 2 norm of the vector with entries equal to the input polynomials substituted at each output root. For each method the residual is divided by the residual calculated for the Simplified G-N method to get a relative residual that will be less sensitive to the scaling of the individual test problems. It also allows for easy comparison with the Simplified G-N method. By definition then this value will be precisely 1.0 for the Simplified G-N method. The three columns report the minimum, arithmetic mean, and maximum of this relative residual among all the convergent test cases.

The next two columns report the minimum and maximum residual calculated for the sample problems. These are not scaled relative to the Simplified G-N result. A smaller value here suggests that the output roots are closer to being roots of the input polynomials. A value less than one suggests that the output roots are closer to being roots of the original system than the input roots.

The *Abs Output Norm* columns report the minimum, mean, and maximum absolute output norm, i.e. the 2-norm of the coefficients of the perturbation functions. A smaller value means the output polynomials have coefficients closer to those of the input polynomials.

The *Rel Output Norm* columns report the minimum, mean, and maximum relative output norm. Values smaller than 1.0 indicate a smaller (better) absolute output norm than the Simplified G-N method.

The *Iter Cnt* column reports the average number of iterations required until convergence is achieved.

## References

- [1] W. Auzinger and H. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. Intern. Conf. on Numerical Math., Intern. Series of Numerical Math.*, 86, pages 12–30. Birkhauser Verlag, Basel, 1988.
- [2] A.-M. Bellido. Construction of iteration functions for the simultaneous computation of the solutions of equations and algebraic systems. *Numerical Algorithms*, 6:313–351, 1994.

- [3] W. S. Brown and J. F. Traub. On Euclid's algorithm and the theory of subresultants. *Journal of the ACM*, (18):505–514, 1971.
- [4] P. Chin, R. M. Corless, and G. F. Corliss. Optimization strategies for the approximate gcd problem. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 228–235. ACM Press, 1998.
- [5] G. E. Collins. Subresultants and reduced polynomial remainder sequences. *Journal of the ACM*, 14(1):128–142, 1967.
- [6] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 195–207, 1995.
- [7] J. Dedieu and M. Shub. Newton and predictor-corrector methods for overdetermined systems of equations. Technical report, IBM Research Division, 1998.
- [8] J. P. Dedieu and M. Shub. Newton's method for overdetermined systems of equations. *Math. Comp.*, 69(231):1099–1115, 2000.
- [9] E. Durand. *Solutions numériques des équations algébriques. Tome I: Équations du type  $F(x) = 0$ ; racines d'un polynôme*. Masson et Cle, Editeurs, Paris, 1960.
- [10] E. Durand. *Solutions numériques des équations algébriques*, volume 1. 1968.
- [11] M. Elkadi, A. Galligo, and T. L. Ba. Approximate GCD of several univariate polynomials with small degree perturbations. *J. Symbolic Comput.*, 47(4):410–421, 2012.
- [12] I. Z. Emiris, A. Galligo, and H. Lombardi. Numerical univariate polynomial GCD. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, pages 323–343, 1996.
- [13] I. Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *J. Pure Appl. Algebra*, 117/118:229–251, 1997. Algorithms for algebra (Eindhoven, 1996).
- [14] A. Frommer. A unified approach to methods for the simultaneous computation of all zeros of generalized polynomials. *Numer. Math.*, 54:105–116, 1988.
- [15] M. Giusti and É. Schost. Solving some overdetermined polynomial systems. In *ISSAC '99*, pages 1–8. ACM, 1999.
- [16] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.
- [17] J. D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Trans. Math. Software*, 38(4):Art. ID 28, 20, 2012.

- [18] M. A. Hitz and E. Kaltofen. Efficient algorithms for computing the nearest polynomial with constrained roots. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (Rostock)*, pages 236–243, New York, 1998. ACM.
- [19] M. A. Hitz, E. Kaltofen, and Y. N. Lakshman. Efficient algorithms for computing the nearest polynomial with a real root and related problems. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, pages 205–212, 1999.
- [20] E. Kaltofen, Z. Yang, and L. Zhi. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *ISSAC 2006*, pages 169–176. ACM, New York, 2006.
- [21] E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a Sylvester matrix. In *Symbolic-numeric computation*, Trends Math., pages 69–83. Birkhäuser, Basel, 2007.
- [22] N. Karmarkar and Y. N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 35–39, 1996.
- [23] N. K. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *Journal of Symbolic Computation*, 26(6):653–666, 1998.
- [24] I. Kerner. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math.*, 8:290–294, 1966.
- [25] B. Li, J. Nie, and L. Zhi. Approximate GCDs of polynomials and sparse SOS relaxations. *Theoret. Comput. Sci.*, 409(2):200–210, 2008.
- [26] H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numer. Math.*, 70(3):311–329, 1995.
- [27] B. Mourrain and O. Ruatta. Relation between roots and coefficients, interpolation and application to system solving. *Journal of Symbolic Computation*, 33(5):679–699, 2002.
- [28] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM Rev.*, 39(2):187–220, 1997.
- [29] N. Rezvani and R. M. Corless. The nearest polynomial with a given zero, revisited. *SIGSAM Bull.*, 39(3):73–79, 2005.
- [30] O. Ruatta. A multivariate Weierstrass iterative rootfinder. In *ISSAC*, London, Ontario, 2001. ACM press.
- [31] O. Ruatta. *Dualité algébrique, structures et applications*. PhD thesis, Université de la Méditerranée, 2002.

- [32] D. Rupperecht. An algorithm for computing certified approximate GCD of  $n$  univariate polynomials. *J. Pure Appl. Algebra*, 139(1-3):255–284, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).
- [33] A. Schönhage. Quasi-gcd computations. *Journal of Complexity*, (1):118–137, 1985.
- [34] H. Sekigawa. The nearest polynomial with a zero in a given domain. *Theoret. Comput. Sci.*, 409(2):282–291, 2008.
- [35] B. Sendov, A. Andreev, and N. Kjuskiev. *Handbook of Numerical Analysis*, volume III, chapter Numerical Solution of Polynomial Equations, pages 628–777. Elsevier, 1994. Solution of Equations in  $\mathbb{R}^n$  (part 2).
- [36] H. J. Stetter. Condition analysis of overdetermined polynomial systems. In E. V. V.G. Ganzha, E.W. Mayr, editor, *Computer Algebra in Scientific Computing - CASC 2000*, pages 345–366. Springer, 2000. <http://www.math.tuwien.ac.at/~stetter/listealg.html>.
- [37] H. J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.
- [38] K. Weierstrass. *Neuer Beweis des Fundamentalsatzes der Algebra*, *Mathematische Werke. III*. Mayer und Mueller, Berlin, 1903.
- [39] J. R. Winkler and J. D. Allan. Structured low rank approximations of the Sylvester resultant matrix for approximate GCDs of Bernstein basis polynomials. *Electron. Trans. Numer. Anal.*, 31:141–155, 2008.
- [40] J. R. Winkler and J. D. Allan. Structured total least norm and approximate GCDs of inexact polynomials. *J. Comput. Appl. Math.*, 215(1):1–13, 2008.
- [41] J. R. Winkler and M. Hasan. A non-linear structure preserving matrix method for the low rank approximation of the Sylvester resultant matrix. *J. Comput. Appl. Math.*, 234(12):3226–3242, 2010.
- [42] J. R. Winkler and M. Hasan. An improved non-linear method for the computation of a structured low rank approximation of the Sylvester resultant matrix. *J. Comput. Appl. Math.*, 237(1):253–268, 2013.
- [43] J. R. Winkler, M. Hasan, and X. Lao. Two methods for the calculation of the degree of an approximate greatest common divisor of two inexact polynomials. *Calcolo*, 49(4):241–267, 2012.
- [44] J. R. Winkler and X. Lao. The calculation of the degree of an approximate greatest common divisor of two polynomials. *J. Comput. Appl. Math.*, 235(6):1587–1603, 2011.
- [45] K. Yokoyama, M. Noro, and T. Takeshima. Solutions of systems of algebraic equations and linear maps on residue class rings. *J. Symbolic Comput.*, 14(4):399–417, 1992.

- [46] Z. Zeng. A method computing multiple roots of inexact polynomials. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 266–272, 2003.
- [47] Z. Zeng. Computing multiple roots of inexact polynomials. *Mathematics of Computation*, (74):869–903, 2005.
- [48] Z. Zeng and B. H. Dayton. The approximate gcd of inexact polynomials. In *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 320–327. ACM Press, 2004.
- [49] L. Zhi. Displacement structure in computing approximate GCD of univariate polynomials. In *Computer mathematics*, volume 10 of *Lecture Notes Ser. Comput.*, pages 288–298. World Sci. Publ., River Edge, NJ, 2003.